e

LEVEL II

12

Anaphora for Limited Domain Systems

Philip J. Hayes

30 July 1981

DEPARTMENT
of
COMPUTER SCIENCE

Carnegie-Mellon University

14 CMU-CS-81-136

6

Anaphora for Limited Domain Systems

10 Philip J. Hayes

11 30 Jul[redacted]81

12 17

15 F33615-78-C-1551, ARPA Order-3597

Accession For	
NTIS GRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification ltr on file	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

Abstract

This paper presents a simple mechanism for the resolution of anaphora in limited domain natural language systems This mechanism provides functionality equivalent to the natural communication mechanism of anaphora as used and understood by people, but without the deep inferencing or cognitive modelling required for full simulation of human performance. The mechanism covers simple pronoun anaphora, and set selection anaphora (e.g. "last one", "one before", "others"). It was developed to provide the most efficient and effective communication between system and user, even if this meant diverging significantly from human performance when this performance was impractical to reproduce. In cases of radical divergence, we were careful to make the behaviour of the mechanism very simple and easy to predict. In this way, the user can either rely on his experience of human performance or his knowledge of the artificial, but simple, substitute to predict the behaviour of the system in response to his inputs, and thus construct his inputs to use the mechanism to best advantage. An algorithmic description of an implemented version of the mechanism is presented. A similar approach to other aspects of man-machine interfaces is recommended as a promising way to address the problem of habitability that still plagues all natural language computer interfaces.

This research was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-78-C-1551. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

To appear in proceedings of 7th Int. Jt. Conf. on Artificial Intelligence, Vancouver, August, 1981.

A

403081

1. Introduction

There has been considerable interest recently in natural language interfaces to interactive computer systems. A primary motivation behind such interfaces is the belief that a user will be able to communicate much more naturally and easily with a computer system if he can do it in his own language, instead of having to learn · special artificial one. However, the ability to parse natural language into an internal formalism is not in itself sufficient to make an interface appear natural and graceful to its user. To provide this impression, an interface must also deal with pragmatic linguistic phenomena, such as ellipsis and anaphora, including those that arise from spontaneous language use, such as grammatically deviant utterances. Mechanisms to deal with some of these phenomena have been developed for many recent natural language interfaces to limited domain systems, including SOPHIE [1], LIFER [6], PAL [7] and PLANES [9], as well as in work by Grosz [4], and by Hayes and Mouradian [5].

In this paper, we present another such mechanism; one to resolve anaphoric reference by pronouns or noun phrases. While anaphora resolution has been provided by several of the systems mentioned above, the mechanism we present deals with some aspects of the phenomena that have not been considered before, and perhaps more importantly, is the product of a distinctive approach to the development of such mechanisms. In developing our anaphora mechanism, we have regarded the anaphora employed by humans as a natural communication mechanism which provides its users with a certain functionality, viz. the ability to identify certain things to their hearers via highly abbreviated descriptions such as pronouns. Our aim has been to provide the users of a computer system with an ***equivalent*** functionality, viz. the ability to identify certain things to the system via similarly abbreviated descriptions. The stress here is on equivalent. In other words, given the two goals of:

1. simulating human performance in the interpretation of anaphora;
2. providing the functionality of anaphora in whatever form will make for the most effective and efficient communication between system and user.

we have chosen the second in preference to the first. Now, one might argue that the best way to achieve goal 2 is, in fact, to achieve goal 1. However, even if this is true, pragmatic considerations rule out preferring 1 over 2 for any interface to be constructed in the near future. As research on the broader problem of anaphora resolution in general natural language (e.g. [2, 3, 8]) has shown, simulation of human performance requires the application of massive amounts of real-world knowledge. Even given the much more constrained worlds, such as data base retrieval or command interaction, that natural language interfaces typically operate in, full simulation of human

understanding of anaphora would require far more information than it is currently feasible to encode in a practical interface.

The fact that we chose goal 2 over goal 1 does not mean that we believe human performance on anaphora should be totally ignored. Indeed, if we can base the performance of our mechanism on the human version, it will be much easier for a human to use the mechanism because he will be able to predict the behaviour of the mechanism by analogy to human behaviour. The only problem is that the system user might over-generalize and predict that the system's anaphora mechanism will perform the same as a human in cases where it cannot. For this reason, any interface system using this mechanism must have good feedback techniques to alert the user to when this might be happening. On the other hand, choosing goal 2 over goal 1 enables us to depart radically from human performance if this turns out to make things simpler for our anaphora mechanism without reducing the functionality available to the user. However, when we do this, we must be careful that it is easy for the user to understand and remember how those aspects of the mechanism will work, since he cannot rely on his human expectations. We can sum this up as follows:

- In those areas in which the mechanism approximates human performance, we can rely on a user's human expectations, so long as we take appropriate precautions against over-generalization.

- In those cases in which the mechanism diverges radically from human performance, the way in which the mechanism operates must be easy to understand and predict.

The anaphora mechanism we present below meets these criteria. Remaining sections of the paper: delimit the range of functionality provided by the mechanism, describe the mechanism from the user's point of view, present algorithmic details of the mechanism, and finally examine some problems that still remain.

2. Aspects of Anaphora to be Covered

This section presents the types of anaphoric references covered by the mechanism we have developed for the resolution of anaphora in limited domain natural language interfaces,

The device of anaphora in human language allows a speaker to identify an entity to his listener without giving a "full" description of that entity, When a speaker describes something anaphorically, the description he gives is not sufficient to distinguish the entity from everything that the listener knows about, but only to distinguish it from a much smaller group of entities that are directly related to the current state of the dialogue or discourse. The resolution of anaphoric references in general natural language has been studied actively for some time now, see Charniak [2] or Sidner [8] for an

introduction to the literature of this area[1]. The work reported here is not presented as a theoretical advance in this area (although the aspect concerning set selections, see below, might have some interest from that point of view), but rather as a piece of applied natural language research in the spirit of the PLANES system of Waltz [9], or the LIFER system of Hendrix [6].

The functionality of anaphora is very useful for computer interfaces (whether based on natural language or not). The full description of, say, a file or a data base item might be quite lengthy, requiring considerable typing, so it would be very convenient for the user of a computer system that dealt with such objects to be able to refer to them, using a (presumably much shorter) anaphoric description, once they had been introduced into the interaction either by him or by the system. The basic functionality we wish to provide the user of an interactive computer interface is, therefore, the ability to refer to system objects without describing them fully, after they have been introduced into the dialogue, and while the dialogue is still "related" to them. A precise definition of what "related" means here must wait until the algorithmic description of our anaphora mechanism given in Section 4. In the meantime, let us examine some examples of the kinds of anaphoric descriptions to be covered. The examples we give are all drawn from the domain of an electronic mail system with which we have been experimenting. We assume the system responds appropriately after each line of the user's input.

One of the simplest and most useful forms of anaphora is the use of pronouns to refer back to some recently mentioned object as in:

Display all messages from Campbell
Delete them
Display all messages to him

Here "them" refers to all the messages from Campbell (that the system knows about), and "him" refers to Campbell. Most such references can be interpreted correctly by the well known rule that such pronouns refer back to the last mentioned object of the same type and number, so "him" refers to the last animate, male, singular object, viz. Campbell, and "them" refers to the last plural object, viz. all the messages from Campbell. The selectional restrictions imposed by the actions or states in which the pronouns participate can also be used to help determine the possible referent, so that, in the example above, "them" should refer to a plural object that is deleteable. These selectional restrictions are very powerful in limited domains. For instance, in our example domain, only

[1]Note that we will not make the careful distinctions between reference, specification, indirect specification, and co-specification employed by Sidner. We will confuse all these relations under the broad term of reference. This lack of precision may be partially excused by the lack of any real world associated with limited domain systems. The real world of such a system is its database.

messages are deleteable, so that "them" in the example above must refer to the last mentioned set of messages. A clearer example of the power of selectional restrictions in limited domains is:

Display the messages from Campbell to Murray
Delete them
Display any messages copied to them

Here, the first input mentions three sets of objects, of types message and person, respectively, but selectional restrictions allow the referents of "them" in the second and third inputs to be properly identified. It is only the leverage available from the strong selectional restrictions typical of limited domains that makes such domains so tractable for natural language processing. Inferences that would normally require much more complicated mechanisms in more general domains can be "compiled into" the selectional restrictions. All limited domain natural language systems depend very heavily on this leverage.

The "last-mentioned" rule should not be used to distinguish between items mentioned in the same input, so for example, in:

Display the messages from Campbell to Murray
Display the messages copied to him

"him" is ambiguous between Campbell and Murray. Even the much more sophisticated focus tracking rules proposed by Sidner [8] for referent determination cannot help in this situation. Of course, a more detailed knowledge of the user's goals might enable another human to resolve the referent of "him" quite easily, but this is precisely the kind of information that there is no simple or efficient way to encode in current interfaces. Our anaphora mechanism should, therefore, report that the referent of "him" is ambiguous between Campbell and Murray, and leave it up to other aspects of the interface to determine by interaction with the user which was really intended. One other point to note is that multiple instances of the same type of item in the same input create an implicitly mentioned set of the union of all the instances, so that if "him" was replaced by "them" in the last example, "them" would refer to the set of Campbell and Murray[2].

Besides straightforward references back to previously mentioned objects, a second very useful type of anaphora that has received less attention in the literature relates to pronouns which select certain elements out of larger sets. These selections may be absolute (the last one, the first three, the second through the fourth), or relative to other elements of the set (the one before, the following two, the others). Examples of the use of these pronouns include:

---

[2]Whether the input would then be a request to display all the messages to Campbell plus all the messages to Murray or only those messages that are to both Campbell and Murray is not a matter for the anaphora mechanism to decide.

Show the headers of the messages from Campbell
Display the last one
Display the one before
Display the first two
Delete the others

Note that "others" refers to all the messages from Campbell except for the first two and the last two, although, as in the case of any destructive operation whose operands were obtained through anaphora, a prudent interface would ask for confirmation of this conclusion. The anaphora algorithm we present below allows the use of such set selection pronouns. In essence, it allows absolute set selection pronouns to select from the last set mentioned that satisfies the selectional restrictions applicable to the pronouns; relative set selection pronouns operate relative to the last selection from the set, or in the case of "others", relative to all the selections that have been made from the initial superset.

In all the preceding examples, we have assumed that the underlying message system maintained a universe of objects of type message and person, so that referents for "Campbell" and "all the messages from Campbell" could be determined by the system. This assumption of a global data base or context against which non-anaphoric descriptions can be resolved is an appropriate one for most limited domain interfaces. In the cases we looked at above there was never any confusion about whether a description should be resolved against the global context or treated as anaphoric references to the local context; pure pronouns are always treated as anaphors, and the other descriptions were clearly non-anaphoric. But consider examples like:

Display the messages from Campbell
Display the ones to Murray

Display the messages from Campbell
Delete the last message

Does "the ones to Murray" mean those messages that are both from Campbell and to Murray or all the messages in the global context that are to Murray? Does "the last message" mean the last one of those from Campbell, or the last one in the global ordering (assuming such an ordering exists)? In general these questions are impossible to answer without knowing why the user is issuing the commands he is issuing, and as we said before, attempting to keep track of such information is impractical for current technology.

To deal with this problem of deciding whether a particular description is global or anaphoric, we have adopted the following simple principle: descriptions that are headed by pronouns will be considered anaphoric, and descriptions headed by nouns will be considered to refer globally. Thus, in the examples above, "the ones to Murray" means the messages that are both from Campbell and to

Murray, while "the last message" means the last message in the global order. This rule will result in behaviour that is sometimes quite different from the behaviour of a human in similar circumstances, but the complexity of the information involved in simulating human performance in this case is too great and a compromise of some sort is unavoidable. The rule we have adopted will correspond to human behaviour more often than not, but most importantly it is simple and results in easily predictable system behaviour. If the user wants to refer to something anaphorically, it is easy enough for him to remember to use a pronoun, while if he wants to give a global description, he need only use the appropriate noun. And if he forgets, the unexpected result will be easily explainable.

3. Presenting the Anaphora Mechanism to the User

Our prime requirement for providing a computer interface with the functional equivalent of a natural communication mechanism like anaphora is that the workings of the mechanism be either directly analogous to human performance or very easy to understand and predict. In this way, the user can either rely on his experience of human performance or his knowledge of the artificial, but simple, substitute to predict the behaviour of the system in response to his inputs, and thus construct his inputs to use the mechanism to best advantage. The following user-level description of our anaphora mechanism demonstrates that this mechanism fulfills this requirement. Note that the more the mechanism diverges from human performance, the more acute the need for simplicity becomes. In the case below, the aspect of the mechanism which displays the greatest divergence from human performance, viz. the radically different treatment of pronouns and nouns, is also the simplest aspect to explain.

1. The interface maintains a current context of all the system objects mentioned by the user or the system.

2. You may use the pronouns, he, she, it, they, and their derived forms, to refer back to objects in the current context. Only descriptions involving pronouns will be interpreted with respect to the current context, descriptions whose head is a noun will be interpreted with respect to the global context in the normal way.

3. The referent of such a pronoun is the last mentioned object that agrees with the pronoun in type (Message, Person, Date, etc.) and number (singular or plural). The input context is considered in determining the type of a pronoun (e.g. in "Display it", "it" is restricted to be Message since the argument to display must be a message).

4. The pronoun, one, may be used:
   a. in noun phrases, just like any other noun known to the system (e.g. "the ones from Smith"),
   b. with the adjectives, last, first, second, third, etc. to refer to an individual member of a larger set (e.g. "the second one", "the last one from Smith").

c. with the adjectives, before, after, to refer to set members on either side of those previously selected (e.g. "the one before").

5. Wherever appropriate, numbers other than one may be used as pronouns in the same way as one (e.g. "the first three from Smith", "the two after").

6. Once some elements have been picked out of a larger set, the pronoun, others, may be used to refer to all remaining elements of the set.

The above explanation relies on the user's understanding of the human treatment of anaphora for those aspects of the mechanism that most closely approximate human performance. However, these aspects are still not exact simulations of human performance, and so the explanation must be supplemented by examples which show how the mechanism actually works, especially when it fails to follow human norms. Such examples and counterexamples provide protection against the user over-generalizing and crediting the system with greater capabilities than it really has.

The following examples, commented as they might be for a user's benefit, were generated through a preliminary implementation of the anaphora mechanism according to the description given in the next section. The implementation is preliminary in the sense that it is not yet connected either to a real message system or to a parser - its input was parsed into internal format by hand. Approximately three times the number of examples given here is necessary to cover all aspects of the mechanism's behaviour in what appears to us sufficient detail for a novice user. However, no reasonable number of examples is likely to familiarize the user with every idiosyncracy of the mechanism, so detailed feedback of the referent choices made by the mechanism will still be essential when the mechanism is actually functioning in an interface.

display the messages from Campbell

Because this description contains no pronouns, it is interpreted with respect to the global context.

2	June 7	Campbell	meeting tommorrow
12	June 12	Campbell	paper details

delete them

"them" refers to the last mentioned set of messages - only messages can be deleted.

Messages: 2, 12 deleted

display any messages to him

the messages description is headed by a noun, and so is interpreted globally, but its subdescription, "him", is a pronoun, and refers to the last mentioned person (either male or female - the system doesn't understand the difference); in this case the person is Campbell.

3	June 7	=>Campbell	re: meeting
8	June 10	=>Campbell	paper deadline
18	June 20	=>Campbell	conference proceedings
20	June 25	=>Campbell	TGIF
25	July 3	=>Campbell	recent bugs
35	July 6	=>Campbell	more bugs

display the fifth one

Single messages are displayed in full format, and multiple messages are displayed as above in header format, so at this point, the system displays the full text of message 25. Message 25 is chosen rather than message 5, because pronouns are interpreted relative to the local rather than the global context.

display the one before

For similar reasons, the system displays message 20, not message 24.

display the one after

The system displays message 35, because the last two selections from the larger set form a contiguous subset, and message 35 is the one directly after that contiguous subset.

delete the others

Messages: 3, 8, 18 deleted

After one or more such selections have been made from a set of objects, "others" refers to all the members of the set that have not yet been selected. The system does not ask for a confirmation here since an undelete operation is available.

display the last one

Message 35 is displayed again. The absolute set selections in any unbroken series of absolute and relative set selections are always interpreted relative to the same superset. This superset, called the current selection superset, will always be the superset from which the first absolute set selection in the sequence was made. Note also that the system cannot take into account that it is being asked to do the same thing over again.

delete the others

Messages: 20, 25 deleted

Whenever, "others" is used, it resets the system's memory of the cumulative subset of objects that have already been selected from the current selection superset, so that only the objects referred to "others" appear to have been already selected. Thus, the previous use of "others" reset the "already selected" subset to be {3, 8, 18}, the intervening reference to message 35 added it to this set, leaving {20, 25} as the referent set for "others".

undelete them

Messages: 20, 25 restored

Pronouns can be used to refer to sets or individuals selected from a larger set through anaphora.

4. An Algorithm for the Anaphora Mechanism

In this section, we give details of the algorithm we have implemented to provide the anaphora mechanism we have been describing. The task of the algorithm is to map already parsed input descriptions into system objects (e.g. to map "the ones from Campbell" into a set of messages). The parsed form of an input description is a structure with the components:

type:	one or a list of domain types that the description may refer to.
number:	singular or plural
pronoun:	true or false, depending on whether the head of the description is a pronoun.
components:	a property list of descriptions of components of the top-level object; the corresponding components of the top-level referent must agree with these component descriptions.
abs-set-sel:	indicates a selection from a larger set, specified by an offset from the start or end of the set.
rel-set-sel:	indicates a selection from a larger set, specified relative to a previous selection from that set.

Thus, "the last one from Campbell" would be represented by:

```
[
 type: message
 number: singular
 pronoun: true
 components: [sender: <representation for "Campbell">]
 abs-set-sel: [start: -1 number: 1]
]
```

The "start" of the abs-set-sel specifies the starting place for the selected subset in the ordered superset, negative numbers count backwards from the end of the set.

The algorithm maintains three structures: CURRENT, NEW, and BACK-UP. CURRENT contains the context established by the last input from the user and the reply to it by the system. BACK-UP contains the residual context from earlier inputs. NEW contains the context built up during the processing of the current input and the reply to it by the system. BACK-UP contains a list of individual objects, and a list of (non-empty, non-singleton) sets of objects. These constitute potential singular and plural referents for pronouns which do not refer to anything in the last exchange between the user and system. The lists will generally contain the last mentioned singular and plural objects of each type. CURRENT and NEW have an identical structure - CURRENT is replaced by NEW after each exchange between the user and system. They contain lists of individuals and sets just as in BACK-UP, and in addition, have a list of *selection records*. A selection record consists of a *superset* - the one selected from, a *current subset* - the subset (possibly a singleton) last selected from it, a *cumulative subset* - the union of all subsets that have been current so far (unless it is reset by

"others" - see below), and a *contiguous subset* containing the union of all subsets produced by the last unbroken sequence of relative set selections, together with the absolutely selected subset the sequence started with.

Referents are determined according to the following algorithm, which uses CURRENT and BACK-UP only.

1. Apply the algorithm recursively to any descriptions in components of the current description. Replace those descriptions by their referents.

2. If pronoun of the current description is false, the referent is the set of all objects in the global context that agree with the components of the description.

3. If there is no abs-set-sel or rel-set-sel, and the description is plural, find all the sets listed in CURRENT that agree with the description in type and components. If there is just one of these, it is the referent; if there is more than one, the reference is ambiguous. If the description has components, the component matching procedure may select a subset of one of the sets in the list. If so, the selected subset will be the referent (or one of them in the case of ambiguity). If no referent is found in CURRENT, repeat the search in the list of sets of BACK-UP[3].

4. If there is no abs-set-sel or rel-set-sel, and the description is singular, perform a search analogous to the one in step 2, but using the lists of individual objects of CURRENT and BACK-UP. If no referent is found this way, perform the search exactly as in step 2, using the sets of CURRENT and BACK-UP. Referents found this way will be ambiguous, unless the component matching selects exactly one element out of exactly one of the sets.

5. If there is an abs-set-sel, the first step is to determine the set to be selected from. This is done by matching the description against the sets in CURRENT, but not BACK-UP, in the same way as in step 2, i.e. as if it were plural and had no set-selection. The only change is that sets which are not in the ordinary list of sets in CURRENT, but are supersets of one of the selection records of CURRENT, are considered first as possible referents. Once this preliminary set has been found, the actual referent is found by applying the set selection to it.

6. If the description has a rel-set-sel, the referent must be determined relative to one of the selection records in CURRENT. The selection record to use is determined by the matching procedure of step 2, but applied only to the supersets of the selection records. Once the selection record has been identified in this way, the referent of the description is determined by either taking the complement of the cumulative subset, in the case of "others", or by taking the elements immediately preceding or following the contiguous subset, in the case of n before or after.

---

[3]Note that searching in CURRENT before searching in BACK-UP helps to resolve pronouns with no inherent type restrictions (like "them") when the input context does not provide sufficient selectional restrictions to determine the type of the pronoun. In such cases, the referent will be just those items in CURRENT of the corresponding number. This provides part of the effect of the focus stack in the model of focus movement developed by Sidner [8], but cannot duplicate the effect of returning to a previous context that occurs when this stack is popped.

This algorithm can, of course, either fail to produce a referent at all, or produce set referents for singular descriptions, or individual referents for plural descriptions. We take the view that such problems should be resolved by the other parts of the interface, possibly in consultation with the user, rather than by the anaphora mechanism itself.

The other important part of the anaphora mechanism concerns the way in which NEW is constructed while the current description is being resolved, and how CURRENT and BACK-UP are modified ready for the next exchange between the user and system. The procedure is as follows:

1. Every referent found during the processing of the current input is added to NEW, as either one of its individuals or one of its sets as appropriate.

2. If a referent is found as a result of an absolute set selection from a set which is not a superset of a selection record in CURRENT, then a new selection record is added to NEW. The superset of this record is the set selected from, the current subset is the referent, the cumulative subset contains the one or more objects involved in the referent, and the contiguous subset is the same as the cumulative subset. Note that the subset selection possibility mentioned in step 2 of the referent determination procedure is also treated as an absolute set selection for the purposes of this step and the next one.

3. If a referent is found as a result of an absolute or relative set selection from a set which is the superset of a selection record in CURRENT, then the selection record is copied from CURRENT to NEW, changing the current subset to the referent just found, and with the following additional changes:
   a. If the selection was absolute, the contiguous subset is the same as the current subset, and the cumulative subset is the union of the current subset and the previous cumulative subset.

   b. If the selection was "others", the cumulative and contiguous subsets are made the same as the current subset.

   c. If the selection was a relative selection different from "others", the cumulative and contiguous subsets are augmented by the elements of the current subset.

4. Any objects or sets of objects mentioned in the system's response are added to NEW in the manner specified in step 1.

5. After the response has been completed and before the next input, CURRENT is replaced by NEW, and all objects in the individuals and sets of NEW are added to BACK-UP. Any objects already in BACK-UP of the same type and number are removed at this time. If this procedure involves adding to CURRENT more than one individual object of the same type, the set of all those individuals is also added to the set lists of BACK-UP and CURRENT.

6. Finally, NEW is made empty in preparation for the next input.

5. Problems with the Anaphora Mechanism

While the anaphora mechanism we have been considering provides much of the functionality of the corresponding natural communication mechanism without any deep inferencing or cognitive modelling. there are still some aspects of its behaviour which neither correspond to human intuition, nor operate straightforwardly enough to make prediction by the user easy. We present below the two such aspects of the algorithm's behaviour that we are currently aware of, together with some thoughts about how to improve them.

Both problems concern the set selection part of the mechanism. The first is the most important and concerns multiple levels of set selection. An example is:

Display the messages from Campbell
Display the ones copied to Smith
Delete the last one

The first message description is resolved globally because it is headed by a noun. The second is resolved as a subset of the messages from Campbell, since it is headed by a pronoun, and that set was the last mentioned set of messages. The question is: which set of messages should "the last one" be resolved against? The current mechanism resolves it against the messages from Campbell because it is a set selection and set selections are always resolved first against currently selected sets. This avoids the problem of developing multiple levels of selected sets, and is not an unreasonable interpretation for the above example, representing an easy to predict choice between what are, from the human point of view, two ambiguous interpretations. Unfortunately, it provides an unintuitive result in the case of:

Display the messages from Campbell
Delete the ones copied to Smith
Undelete the last one

The problem here is that the user is able to apply knowledge (about only undeleting messages that have been already deleted) that cannot be encoded into the selectional restrictions of the domain, and so is unavailable to our anaphora mechanism.

There are two potential solutions to the problem: one is to allow multiple levels of selected sets and make the anaphora mechanism generate ambiguous (to it) alternative referents in such situations, leaving the resolution up to a more knowledge-intensive part of the system. The second alternative, more in keeping with the spirit of our enterprise, is to document this aspect of the system's behaviour very carefully, and expect the user to learn (with the support of the feedback system) to live with it. To make make this second alternative more palatable to the user, it would be possible to augment the anaphora mechanism to allow the user to specify the set to be selected from explicitly, as in:

Display the messages from Campbell
Delete the ones copied to Smith
Undelete the last of them

By the rules of the existing system, "them" refers to the messages that are from Campbell and copied to Smith, and a possessive construction of this form could conventionally be interpreted to indicate the set to be selected from.

A second problem with the set selection mechanism involves the treatment of "others". In:

Display the messages from Campbell
Display the last one
Display the one before
Delete it
Delete the others

there is a problem about whether "the others" includes the last of the messages from Campbell. The present algorithm would say that it does, but a human would lean towards saying that it does not. Note that there would be no doubt that it did if the fourth input "Delete it" were omitted. One solution would be to change the accumulated subset of a selection record to be its current subset if the current subset were ever referred to directly by a pronoun.

Neither of these problems can be resolved satisfactorily without testing potential solutions through use of the anaphora mechanism in a real interface. Such testing will probably also turn up other problems that we have not yet considered. We expect to carry out such tests in the near future. Another interesting experiment that we are considering would involve giving the anaphora mechanism the ability to explain why it made a referent selection the way it did. Such an explanation facility might make it much easier for a user to learn to live within the limits of the mechanism.

6. Conclusion

This paper has presented a simple mechanism for the resolution of anaphora in limited domain natural language systems. For such domains, this mechanism provides functionality equivalent to the natural communication mechanism of anaphora as used and understood by people, but without the deep inferencing or cognitive modelling required for full simulation of human performance. The mechanism covers simple pronoun anaphora, and set selection anaphora (e.g. "last one", "one before", "others"). It was developed to provide the most efficient and effective communication between system and user, even if this meant diverging significantly from human performance when this performance was impractical to reproduce.

We believe the same approach can be used to develop similar algorithms for other natural communication mechanisms such as those involved in the resolution of ellipsis or the recognition of

ungrammatical input, and we suggest this general approach as a way to address the problem of habitability, first raised by Watt [10], that has long been a major stumbling blcck for natural language computer interfaces.

References

1. Brown, J. S. and Burton, R. R. Multiple Representations of Knowledge for Tutorial Reasoning. In *Representation and Understanding*, Bobrow, D. G. and Collins, A., Ed.,Academic Press, New York, 1975, pp. 311-349.

2. Charniak, E. C. Toward a Model of Children's Story Comprehension. TR-266, MIT AI Lab, Cambridge, Mass., 1972.

3. Cullingford, R. *Script Application: Computer Understanding of Newspaper Stories*. Ph.D. Th., Computer Science Dept., Yale University, 1978.

4. Grosz, B. J. The Representation and Use of Focus in a System for Understanding Dialogues. Proc. Fifth Int. Jt. Conf. on Artificial Intelligence, MIT, 1977, pp. 67-76.

5. Hayes, P. J. and Mouradian, G. V. Flexible Parsing. Proc. of 18th Annual Meeting of the Assoc. for Comput. Ling., Philadelphia, June, 1980, pp. 97-103.

6. Hendrix, G. G. Human Engineering for Applied Natural Language Processing. Proc. Fifth Int. Jt. Conf. on Artificial Intelligence, MIT, 1977, pp. 183-191.

7. Sidner, C. L. A Progress Report on the Discourse and Reference Components of PAL. A. I. Memo. 468, MIT A. I. Lab., 1978.

8. Sidner, C. L. Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse. TR 537, MIT AI Lab, Cambridge, Mass., 1979.

9. Waltz, D. L. "An English Language Question Answering System for a Large Relational Data Base." *Comm. ACM 21*, 7 (1978), 526-539.

10. Watt, W. C. "Habitability." *American Documentation 19*, 3 (July 1968), 338-351.